BUILDING SASO WARGAMING SIMULATIONS WITHOUT PROGRAMMERS

Alexander Davis & Ryan Houlette
Stottler Henke Associates, Inc.
San Mateo, CA

ABSTRACT

We have designed and prototyped a new software tool that will permit military planners to rapidly create wargaming systems customized for specific SASO missions without the assistance of a programmer. This tool (KAGES) possesses two major components: the authoring tool and the knowledge representation engine.

The authoring component provides an intelligent, intuitive graphical user interface that can guide the user through the knowledge acquisition (KA) and simulation authoring process. By manipulating a palette of objects on a "mission canvas," the user specifies the entities and domain knowledge necessary to fully describe a mission. KAGES is not simply a visual authoring tool, however. It collaborates with the user during authoring, drawing upon its built-in knowledge engineering expertise to extract the relevant information from the user and encode it. To help the user leverage the experience of past planners, KAGES also maintains a database of previously encoded domain knowledge from which it can dynamically retrieve and adapt elements to fit the current situational context. Of course, the system also allows advanced users to deactivate the intelligent assistance features and directly author missions in the underlying representation for maximum flexibility.

In order to handle the complex data produced by the user interface, KAGES has at its core a knowledge representation engine designed for the codification of SASO domain knowledge. It is capable of managing all of the rules, facts, constraints, entities, and other elements that are pertinent to a particular mission, starting with METT-TP (Mission, Enemy, Troops, Terrain, Time, and Politics) and ranging all the way to social and cultural factors. The engine includes a compiler that can automatically generate wargaming scenarios from its internal knowledge structures, so that once a mission has been specified in KAGES, it can immediately be run as a simulation.

ABOUT THE AUTHORS

Mr. Houlette is a project manager and lead software engineer in the San Mateo, CA office of Stottler Henke Associates. He holds an M.S. in Computer Science (Artificial Intelligence) from Stanford University. His primary interests lie in the areas of intelligent interfaces, autonomous agents, and automated generation of simulations and interactive worlds. During his stay at SHAI, he has participated in the development of a wide range of AI systems. He is currently the co-lead software engineer on a project for the Air Force to develop an AI engine and visual authoring tool that will allow non-programmers to specify the behavior of entities within an arbitrary simulation. Mr. Houlette is also currently leading a project to develop a mixed-initiative scheduling system that will include as a core component a rich capacity for human interaction and collaboration. Previous work includes a prototype of an intelligent agent system for distributed simulation and a prototype graphical decision support system for the placement of acoustic ground sensors.

Mr. Davis is a project manager and software engineer in SHAI's San Mateo, California office and has extensive experience in the application of case-based reasoning and other artificial intelligence technologies to the domain of military simulation. He has an M.S. from the University of New York at Buffalo. Mr. Davis was the project manager for the Automated Flight Test Engineering System (AFTES) project. AFTES involved the extraction of knowledge from structured text documents, allowing new flight test plans and reports to be generated based on the new situation input by the flight test engineer. He was also lead knowledge engineer and lead software engineer on the SH-60R OA/OMIES project. This project involved the creation of an adaptive, intelligent user interface system that enhances sensor employment and target classification by assessing the effective expertise of an operator, including dynamic cognitive capabilities such as situation awareness and information overload, and intelligently enhances the operator-machine interface accordingly. The system addresses the operator's training level, individual cognitive style, and particularly past performance under particular sets of operating conditions.

Report Documentation Page	*Form Approved* *OMB No. 0704-0188*

Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.

1. REPORT DATE **2006**	2. REPORT TYPE	3. DATES COVERED **00-00-2006 to 00-00-2006**

4. TITLE AND SUBTITLE **Building SASO Wargaming Simulations Without Programmers**	5a. CONTRACT NUMBER
	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Stottler Henke Associates Inc,951 Mariner's Island Blvd Suite 300,San Mateo,CA,94404**	8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
The original document contains color images.

14. ABSTRACT

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES **10**	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**			

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std Z39-18

BUILDING SASO WARGAMING SIMULATIONS WITHOUT PROGRAMMERS

Alexander Davis & Ryan Houlette
Stottler Henke Associates, Inc.
San Mateo, CA

INTRODUCTION

The changing face of modern military action has engendered an evolution of terms: Low Intensity Conflict became Operations Other Than War, which became Stability and Support Operations (SASO). The sequence marks a migration of the nature of operations away from purely traditional warfare, and into broad and heterogeneous domains such as peacekeeping, counterterrorism, arms monitoring, and a host of other activities with only intermittent resemblance to outright war. The outcomes of these modern operations are no less crucial, however, and the armed forces involved in SASO will continue to rely heavily on training and course of action analysis through computer simulation. Subsequently such simulations must now consider as significant behavioral factors a far wider field of knowledge than for mere force-on-force: cultural differences among populations, shifting goals and allegiances, unconventional groups of combatants and noncombatants. The terrorist threat is a prime example: its constituents can arise from any sector of society, its motivations draw from such disparate sources as the religious and psychosocial, its goals may be completely free of familiar tactical considerations, and its methods can be grossly asymmetric in devoting very few agents to bringing about massive damage and casualty.

The technical challenges of developing simulations sensitive to the breadth of SASO considerations are formidable. Relevant entities range from federations of countries to individuals, including organizations, populations, forces, factions, and leaders. They must serve a variety of roles, which may shift over time, as parts of different aggregates. Relations between entities, and the characterization of their simulated environment, must include among other factors the psychological, social, political, historical, and economic. Furthermore, any such attribute may be probabilistic and uncertain. All of this new knowledge is ultimately significant only to the extent that the behavior of simulated entities is meaningfully sensitive to it, producing outcomes as widely productive and finely constrained as real life situations, with some useful resolution; the expression of behavior must also include such tasks as monitoring and interdiction, and noncombatant activities that may or may not be threatening. And last, but possibly foremost, is the problem of knowledge acquisition. Substantive knowledge in SASO operations may arise only in the field, and its form may be less easily anticipated by knowledge engineers than in current systems. This requires a flexible acquisition, representation, and execution framework.

THE KAGES SYSTEM

Approaching the problem as one of knowledge acquisition (KA) with users who are experts in subject matter but not in computer or knowledge engineering, and then considering the subsequent implications for knowledge representation and scenario simulation, has proven useful. We share this ambitious objective with a variety of researchers, including the Defense Advanced Research Project Agency's extensive Rapid Knowledge Formation (RKF) project. Our effort differs in immediacy, designed to arrive at a fieldable system in the short term, partly by acceptance of domain dependency but with promise for expansion into an extensive system. The solution amounts to an automated collaborator in the KA process: an intelligent system that employs its knowledge of the SASO domain, KA, and representational requirements for simulation, along with what it observes of the user's knowledge base and preferred level of interaction, in order to balance flexibility with tractability.

In development of the Knowledge Acquisition for Gaming Environment: SASO (KAGES) system, the following principles applied:
-   input must be structured in natural ways
-   unnecessary information must be hidden, and displayed information simplified wherever possible
-   the user must always be provided with context in terms of both the SASO scenario domain and what needs to be done in the KA process
-   KA can be incremental, unordered, and incomplete

The core of KAGES that satisfies these principles is an adaptive collaborator, which employs awareness of the KA process, the SASO domain, and the user. KA process knowledge is incorporated into a set of plans, and into the acquisition manager which follows these plans adaptively in concert with the user's initiative.

SASO domain context is provided through the reuse of knowledge, employing case-based reasoning to adapt past SASO scenarios—and past acquisition episodes—to the user's current task, through the execution of inference modules that attempt to fill in missing information formulaically, and through templates that characterize broadly various SASO mission types. User awareness employs user modeling techniques drawn from adaptive training applications.

WHAT IT DOES

As stated above, the goal of our system is to allow military planners and analysts to encode their mission domain knowledge into a SASO wargaming simulation without the assistance of a programmer or knowledge engineer. Within this overarching purpose, KAGES' functionality can be divided into three main areas: knowledge acquisition, knowledge representation, and scenario generation.

Knowledge Acquisition

The knowledge acquisition component of KAGES is designed to guide the military analyst through the process of specifying a complex SASO scenario. It consists of an intelligent authoring tool that collaborates with the user as a knowledge engineer would, by gauging the desired level of guidance, presenting varied graphical forms of expressing information that appeal to the user's perspectives, and ensuring that the process approaches the goal of a valid executable scenario. If the user expresses interest, more powerful (and complicated) interfaces can be provided. When the process stalls, KAGES can change its approach. For instance, if the user is having trouble identifying relevant entities, the system might switch to behavior expression; determining different situations to be acted upon might suggest what entities bring those situations about. Alternatively, KAGES could search its SASO ontology for groups that the user might not have considered, like political groups or information operators, and ask if they are relevant. More substantive suggestions can arise from the system's adaptations of previous knowledge that are discovered as relevant. We discuss each of these major features in the following sections.

Intuitive Visual Authoring

Most knowledge entry in KAGES is visual, taking place on a "mission canvas" where objects are created and manipulated. The canvas can represent a terrain map, an organizational chart, a behavior diagram, or various other specialized display formats. Standard UI idioms allow clicking and dragging to establish relations, double-clicking to examine individual

objects, and context menus to access available manipulations of an object. Such UI gestures are interpreted intelligently depending on the types of objects involved, possibly querying the user further to specify the meaning of the action. Beside the canvas are other display areas that show object attributes, palettes for the creation of new objects, or various view configurations such as selective display of terrain overlays. These areas of the editor can appear and disappear according to the level of interaction desired by the user, or depending on what types of objects or viewing modes are selected.
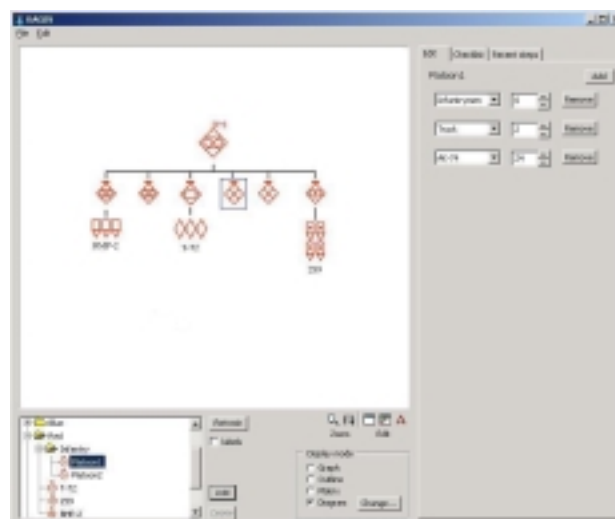

Figure 1 Editor: Red Threat Organization

The accompanying figures demonstrate uses of the intelligent visual editor, including three primary visual editing modes: a specialized threat organization mode (Figure 1), a map editor allowing free sketching of overlays (Figure 2), and a behavior editor (Figure 3). Each consists of objects that are directly manipulable.
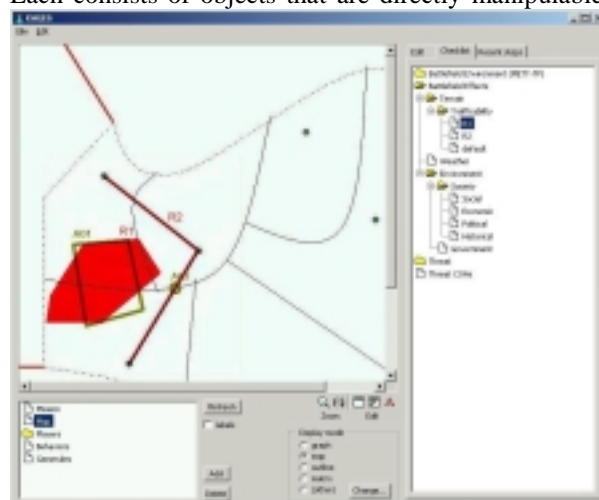

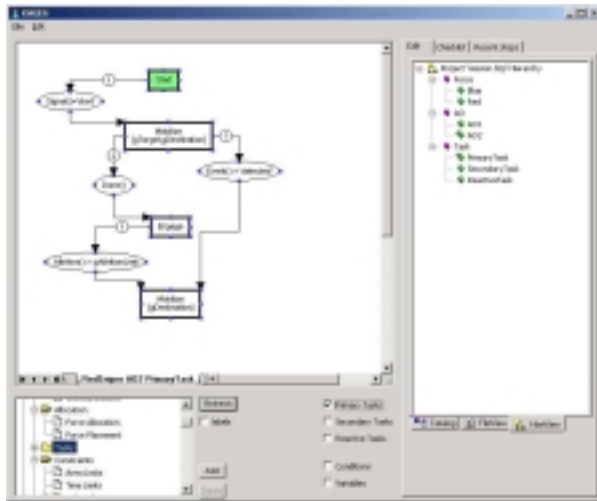**Figure 2 Editor: AO and Trafficability Overlays**

Figure 3 Editor: Red Sniper Behavior

In the first, knowledge about force compositions is expressed. Each unit can be broken down to whatever level of detail is necessary to express behavior. In this case, a platoon and its equipment are specified. The second example shows two sets of overlays simultaneously, one designating AOs and another for specified regions of trafficability. Having drawn these regions, the user can edit them to assign levels of trafficability to each. Entity behaviors can be made sensitive to those regions, and the regions can be transmitted to the simulation if such knowledge is supported in it. The third figure illustrates our entity behavior representation, which uses a flowchart-like diagram to depict complex, nested sequences of decisions and actions.

Multi-Modal, Multiple Level-of-Detail Presentation

The visual editor is the centerpiece of the user's experience with KAGES, but it is not a static form. During the course of its collaboration with the user, the system adjusts the editor's appearance and presentation style, showing or hiding scenario elements according to the skill level and preferences of the user. A novice user, for example, may see a high-level view of entity behaviors, where only the critical decisions and action paths are shown, while an expert user would have direct access to all behavior details. The system's goal is to provide the user with sufficient context for her to be able to specify a mission model *without* overwhelming her with irrelevant information. In addition, the user can choose to manually modify the detail level of the display, providing less detail if the display is too cluttered, or more detail if resolution is too low to allow full expression of the knowledge.

While the visual editor's ability to display multiple levels of detail is a powerful tool for accommodating a wide variety of users, KAGES is also capable of adapting its presentation more drastically. Rather than having a single fixed knowledge-entry interface for each type of knowledge to be acquired, KAGES allows for multiple overlapping knowledge-acquisition modalities. For each knowledge element being elicited from the user, the system determines the most appropriate presentation modality to use. These modalities tend to be familiar methods of eliciting knowledge, such as doctrinal diagrams, maps and overlays, flowcharts, checklists, organizational charts, association matrices, and so on. By using already-existing input formats, the system can leverage the user's existing training and intuitions and minimize the mistakes made due to miscommunications.

It is worth noting that the presentation modalities employed by KAGES may vary widely in form and behavior. Some modalities may be very simple – perhaps no more than a few dialogue windows that prompt the user to enter a few pieces of information – while others may be quite complex, involving sophisticated logic, elaborate diagrams, or graphics-intensive displays.

For example, the user may have specified a set of behaviors that reference the attitudes of various scenario groups toward one another. KAGES recognizes that these relationships have not been expressed, and launches an attitude relationship "wizard" to elicit the knowledge. The wizard will guide the user through selecting groups or individuals of interest, based on the existing scenario or on new information that the user enters at that time. Then it will ask the user to identify the relevant set of relationships given what is known to the SASO ontology, or to author new ones as before. Finally it will present an interactive association matrix for specifying each relationship.

Intelligent Automated Guidance

While the visual authoring system is capable of a wide variety of adaptations and presentation styles, these alone are not enough to enable users with no knowledge engineering skills to encode their mission domain knowledge. Such users need assistance to navigate successfully through the knowledge acquisition process and the complexities of a large knowledge base. The authoring system thus incorporates an intelligent guidance facility that attempts to automate the expertise of the knowledge engineer. KAGES contains a built-in model of the knowledge-acquisition process, and it uses this model to drive a collaborative, mixed-initiative interaction with the user. The level of automation exhibited by the system is determined by the skill level of the user.

Throughout the knowledge acquisition session, KAGES' guidance facility keeps track of the status of the emerging mission knowledge base, noting areas of incompleteness as well as inconsistencies in the model. It then prompts the user (according to one of its various guidance plans) to fill in the gaps or correct the errors in the knowledge base. Collaboration will often consist of an iterative deepening of scenario knowledge, in which the user first defines a simple executable scenario, and then goes on to provide further detail. In fact this is one of various general authoring plans employed by the system; another is a formal IPB process. Each such plan consists of a hierarchy of subplans, and the goal of completing any of these various plans guides KAGES' behavior. The power user is able to view the plans themselves as a sort of authoring checklist, and even modify them to a degree.

While KAGES generally selects the appropriate level of collaboration for the current user, the user can also manipulate the mode of collaboration directly in a variety of ways, through a ubiquitous collaboration control window. One option is a simple "Guide Me" request, which prompts the system to analyze the user's current activity, and determine what needs to be done next. This can consist of an interview process to fill out an incomplete knowledge element, navigation to a subsequent step, or identification of inconsistencies or assumptions in the current domain model. In general, the user is able to tweak the collaboration performed by KAGES to the desired level.

Knowledge Reuse

KAGES constrains the general knowledge acquisition problem through built-in knowledge of the SASO domain. Two forms of this knowledge are templates and cases. Templates are partially populated sets of generic knowledge pertaining to a particular situation. For instance, when the user selects an "Interdiction" scenario, the system can provide default Blue and Red behaviors, and also plan to query the user about locations and the nature of the object of interdiction. Templates exist on smaller scales, for example as sets of reactive behaviors that can be added to a primary behavior. Cases are an entirely different form of help, providing fleshed-out scenarios from past authoring. If

the system determines that such a past case is sufficiently similar to the user's current authoring, it adapts the case as much as possible to the current scenario and presents it to the user for possible inclusion in the domain model. Reusing large portions of past similar scenarios will increase greatly the efficiency of the knowledge acquisition process.

Knowledge Representation

The knowledge representation component of the system is designed to store the set of entities, attributes, relationships, behaviors, constraints, objectives, map data, and other elements of SASO mission knowledge produced by the authoring tool, ranging from conventional METT-T to political and cultural factors. All of this knowledge can be expressed at an arbitrary level of resolution, allowing the user to include such entities as entire populations or individual leaders. Domain models developed by the user are validated based on incorporated SASO knowledge, and when complete can immediately be compiled into executable scenarios.

Scenario Generation

Behaviors are authored in KAGES with the ultimate goal of seeing them played out in a simulation. Given a simulation engine, KAGES can be integrated in two ways. First, KAGES can output a scenario complete with behaviors in some format that a simulation can understand; this separates the authoring system from simulation entirely. Second, KAGES could use its own behavior execution engine in conjunction with the simulation, operating through a software interface. Either method will result in an end-to-end authoring and simulation system, in which the user can repeatedly modify and execute scenarios as necessary.

SYSTEM ARCHITECTURE

The high-level architecture for our system comprises eight major components (see Figure 4). We discuss each of these below.
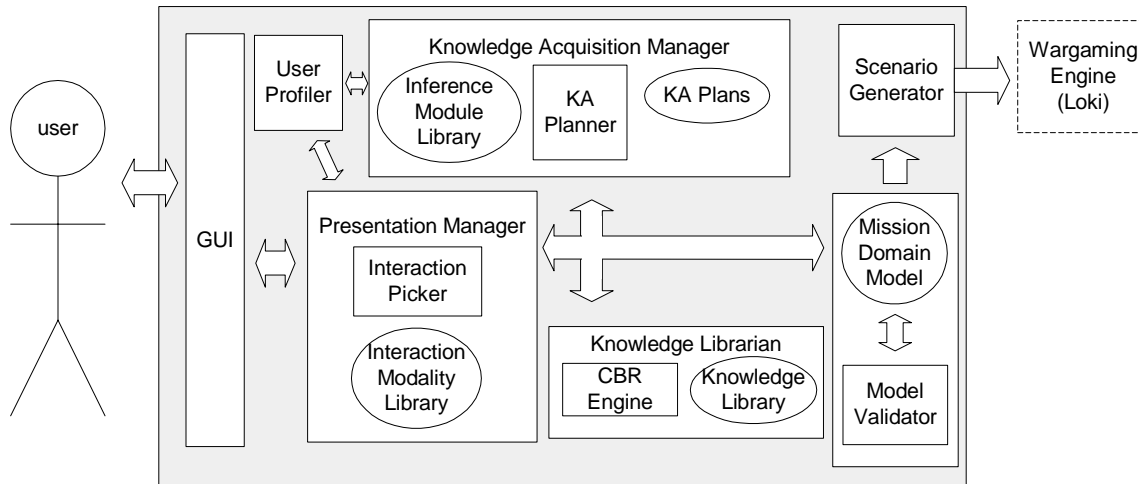
Figure 4 High-Level Architecture

Graphical User Interface (GUI)

As described in the previous section, the user interacts with KAGES via an adaptive graphical user interface (GUI). This interface provides a "toolbox" of visual interaction components that can be dynamically assembled by the system to construct the most appropriate user interface for the current task and user. Because the GUI is highly modular, it is also quite extensible: new interaction components to support new types of knowledge entry can be easily added.

The set of interaction components includes a full set of common elements such as tree-hierarchies, property sheets, wizard dialogs, pop-up menus, drop-down menus, and toolbars. In addition, it includes a variety of more specialized components, such as a map overlay editor, an entity relationship editor, and an entity behavior editor. The map overlay editor allows the user to import standard-format terrain maps and annotate them with overlays describing salient scenario features: unit locations, movement corridors, terrain features, etc. These overlay displays are very similar to the overlays currently in use by the Army, though here they are backed by knowledge structures that permit KAGES to translate them into a format suitable for computational processing.

The entity relationship editor enables the user to visually specify the various entities of interest (e.g., red forces, blue forces, local populations, media) in the scenario and the relationships between them. This editor has many of the features of a modern drawing program like Adobe Illustrator or Visio – a tool palette, a "drawing" canvas, property editors for the currently-selected entity object, and so on – except that the visual objects represent knowledge elements in the domain model. Each type of entity object has logic associated with it that fully describe how it reacts to the user's mouse gestures and how it interacts with the other types of objects. Thus, different entities can intelligently respond to the same user action in the way that is most appropriate to their type.

The entity behavior editor is another visual authoring component, but its purpose is to allow the user to define the behaviors followed by the entities in the scenario. This editor is based upon our BrainFrame visual behavior authoring technology (previously developed for the Air Force), and it permits the user to intuitively specify complex, conditional sequences of actions using a flowchart-like representation. It supports compositional hierarchies of behaviors so that complicated behaviors can be broken down into simpler, more easily-defined units.

The GUI also supports a variety of different intervention modalities that KAGES can use to provide advice and guidance to the user. Dialog boxes can be popped up to prompt the user for information, objects in the editor can be highlighted or pointed to, and interactive help tutorials can be invoked. The GUI can even manipulate its own interface (e.g., clicking on buttons, dragging objects, selecting menu items) to demonstrate how to perform certain tasks to the user.

User Profiler

In order for the system to adapt itself to the needs of the user, it must have some idea about what those needs *are*. Determining those needs is the job of the User Profiler module, which draws upon multiple data sources to create a constantly-updated profile of the current user. Its primary input is the user's own actions, which it monitors for both implicit and explicit

preference information. Implicit preferences are embedded in the user's interaction with the interface – for example, if the user never uses the entity hierarchy tool on the interface, perhaps he or she finds it awkward or difficult to understand. Explicit preferences are directly stated by the user – as, for example, when the user clicks "Do not show me this message again" in an information dialog box, or when the user manually sets the level of guidance desired. The User Profiler may also draw upon summary information provided by the Knowledge Acquisition Manager to evaluate the suitability of the current knowledge acquisition style for the user.

As the User Profiler collects information about the user from the rest of the system, it processes it (using user-modeling technology borrowed from intelligent tutoring system research) and generates a detailed user profile. This profile can be queried by other system modules, which can then adjust their own activity to better suit the user's perceived needs. The result is that a novice user will have a qualitatively different interaction with the system than an expert user – generally speaking, he or she will receive more step-by-step guidance and less low-level detail.

Knowledge Acquisition Manager

The Knowledge Acquisition (KA) Manager is the "brain" of KAGES. It is responsible for orchestrating the behavior of the system and providing the user with consistent, coherent guidance through the KA process. The KA Manager is goal-directed and methodical; it enables KAGES to plan and carry out a sequence of steps to attain its knowledge acquisition objectives rather than simply reacting to each user action on an individual, short-term basis. Its plan-driven nature affords a great deal of flexibility to the KA Manager: it can interrupt the current plan in order to perform an unrelated task and then smoothly pick up where it left off, and it can also abandon an ineffective plan to try a completely different approach.

KA Planner

To enable this kind of intelligent, goal-directed behavior, the Knowledge Acquisition Manager is built upon a hierarchical partial planner. This type of automated planner generates plans according to a very natural strategy of progressive plan refinement whereby a high-level abstract plan is selected first and then broken down into more and more detailed subplans, until finally a plan consisting of low-level performable actions (e.g., "Prompt user to input the terrain for the mission site") has been created. There are obviously often many ways to break down a given plan step; accordingly, the planner possesses a plan library of

possible decompositions from which it automatically selects the one that is most appropriate for the current situation (labeled "KA Plans" in the diagram).

The above-described planner drives the decisions made and actions chosen by the KA Manager. Taking as its inputs the current state of interaction with the user and the contents of the mission domain model, the planner devises a collaboration plan that details what actions KAGES will take and what kinds of responses are expected from the user. These plans may have conditional components to deal with different paths of user action. Once the planner has created a plan, the KA Manager then carries it out step by step. The plan is constantly monitored during its execution to see if the user-system interaction has deviated from the path prescribed in the plan; if it has, the planner will rapidly replan to address the new situation.

Inference Module Library

To be able to generate a runnable wargaming scenario from the information entered by the user, KAGES must have a completely-specified model stored in the Mission Domain Model. In most cases, however, it is highly undesirable to force the user to fully specify every detail of a mission, because it can be both tedious and difficult. The KA Manager thus does not attempt to extract the full mission specification from the user but instead draws upon its library of *inference modules* to fill in the blanks in the Mission Domain Model. These inference modules contain knowledge about the SASO domain, encoded in the form of templates (e.g. common courses of action, standard mission characteristics) and inference rules. Each module is capable of taking an incompletely-specified knowledge element related to its area of expertise, combining that element with its own stored domain knowledge, and producing a new and more complete knowledge element. Based on the kinds of input that are being provided by the user, the KA Manager chooses the set of inference modules needed to fill in the domain model to the desired level of detail. The KA Manager maintains a distinction between knowledge produced by inference, and knowledge specified entirely by the user, to guide further refinement of the model.

Presentation Manager

In KAGES, the decision about <u>what</u> knowledge to ask the user for is separated from the decision about <u>how</u> to ask for it. The KA Manager is responsible for the former, while the latter devolves upon the Presentation Manager module. Once the KA Manager has decided what the current knowledge acquisition goal is to be, it informs the Presentation Manager. This knowledge acquisition goal is stated abstractly: "Ask the user to

specify population distributions for Region X," rather than the more explicit "Display a pie chart for Region X allowing the user to specify the population distribution by sizing the pie wedges." Once the Presentation Manager is notified of this goal, it examines its library of possible interaction modalities and selects an appropriate one (generally drawing upon input from the User Profiler as well). The Presentation Manager then instructs the GUI to display that interaction, waits for the results, and returns the resulting information to the KA Manager.

Note that multiple interaction modalities may be active at once; thus, the Presentation Manager is equipped with algorithms to detect potential conflicts between them. Moreover, many modalities may consist of simplification or abstraction upon the knowledge, leaving gaps in the actual representation. The KA Manager can then compensate by specifically requesting the remaining knowledge, or by appealing to an inference module. An example of this situation arises with the graph-based relationship editor. If it were used to show every entity and every relationship, the result would be an enormous, incomprehensible tangle; therefore every practical use will involve filtering for specific entities and relations between them. The Presentation Manager is capable of both production of this simplified graph, and translating the user's modification of it back to the original representation.

The interaction modalities themselves may vary greatly in their nature. They range from very simple dialogue boxes that request one or two pieces of data to sophisticated, multi-stage interactive "smart wizards" that guide the user through the full specification of some part of the mission. Modalities may often resemble forms of doctrinal knowledge: building link diagrams, plotting event pattern analysis, filling out association matrices, or drawing a wide variety of overlays, as examples. In the absence of a specialized modality, the Presentation Manager can also provide direct access to data representation, for the power user. This is the mode in which the fundamental SASO ontology is modified, and almost all knowledge content is accessible with it.

Knowledge Librarian

The Knowledge Librarian is charged with the duty of helping the user to find and reuse relevant knowledge from previous KA sessions. It is built upon a case-based reasoning (CBR) engine that maintains a library (also known as a *case-base*), consisting of every knowledge element ever created by a user along with its associated contextual information. Included in this

contextual information are, among other things, the date of the element's creation and various details of the mission for which the element was created. This library serves as a repository of mission domain knowledge and experience that grows with each new user interaction.

To make use of this repository, the Librarian incorporates a specialized knowledge retrieval algorithm that can quickly search the library to find past knowledge elements that are similar (both structurally and in terms of mission context) to elements that the user is currently creating. Once the most similar old element has been retrieved, the Librarian adapts it to fit the present context, updating dates, locations, and other attributes so that they match up with the corresponding features of the element-in-progress. The Librarian then informs the KA Manager that it is able to complete some of the user's partially constructed knowledge elements. If the KA Manager deems this to be acceptable in the context of the current knowledge acquisition plan, it will request that the Librarian perform the offered completion; otherwise, the Librarian discards the adapted knowledge element and continues its ongoing retrieval process.

Mission Domain Model/Model Validator

The mission domain model is the central knowledge base that stores the contextual domain knowledge entered by the user. This knowledge is maintained in a special internal format that is designed to facilitate automated processing and inference, with particular emphasis on compilation into wargaming scenarios. As the user manipulates domain knowledge elements in the GUI, the mission domain model automatically translates the user's changes into the corresponding internal representations. The knowledge base engine itself is optimized for compact data storage and fast knowledge retrieval to ensure that KAGES is quick and responsive. (See the discussion on knowledge representation in the Phase I Results section of this proposal for more information on the characteristics of the internal knowledge model.)

Closely integrated with the mission domain model is the Model Validator. This module is designed to catch gaps and conflicts in the domain model that may go unnoticed by the KA Manager. The Model Validator relies upon a set of model validation rules defined by an expert human knowledge engineer to catch errors and missing information in the domain model. Each time a knowledge element is added, changed, or deleted, the Model Validator evaluates its rules to check for newly introduced problems. If a rule *does* fire, indicating a possible error in the model, then the Model Validator

will notify the KA Manager about the problem (and may include as well suggestions about possible resolutions). The KA Manager will then adjust its plan to incorporate model correction tasks. For minor errors, this correction may be automatic; in most cases, however, KAGES will walk the user through the process of fixing the problem.

Scenario Generator

Once the user has specified a mission domain model, KAGES can then translate that model into a scenario suitable for execution in a SASO gaming system. This translation capability is built into the Scenario Generator module, which uses mapping rules, code libraries, and game rule templates to compile KAGES' internal knowledge elements into the equivalent rules, configuration data, units, resources, terrain, and mission details needed by the wargame. The Scenario Generator will draw upon the automated scenario generation capability that SHAI has developed in the past for Navy training simulations. Note that multiple Scenario Generator modules are possible, one for each gaming system that KAGES should support.

Wargaming Engine

While KAGES, using the appropriate Scenario Generator, could be integrated with a variety of different wargaming systems, we will provide a default system based upon our LOKI behavior engine. This engine, which was developed to provide pluggable A.I.-controlled entities in Air Force simulations, is fast, powerful, and easily integrated with a broad class of simulators (though for our purposes, we may use the LOKI engine itself as the simulator, implementing the gaming rules as a set of interacting entities). In addition, the LOKI behavior description language is well understood and will provide an excellent testbed for the domain knowledge-to-wargame scenario translation process.

RELATED WORK

In the area of decision support tools for course of action analysis (COAA), several systems of note have been developed, including the Army Research Lab's FOX-GA (Hayes & Schlabach 1998), CECOM's Course of Action Display & Evaluation Tool (CADET), the Course of Action Selection Tool (COAST), and the Consequence Analysis Tool Set (CATS). While these systems are instructive, neither of them is designed to handle the kinds of small-scale SASO missions that the Army is faced with today. In addition, these tools require the involvement of a programmer or knowledge engineer to customize them for new missions. They do not provide the collaborative knowledge acquisition and custom wargame authoring capability that the KAGES system offers.

Other systems have been developed or adapted to address the SASO knowledge domain, such as Aetheling's NationLab system and Deployable Exercise System (DEXES), and SAIC's Situational Influence Assessment Module (SIAM). While breaking some ground in the engineering of SASO knowledge, they do not support simulation for fine-grained COAA. DEXES, for example, produced outcome predictions in the form of global societal effects. These tools provide insight into the wider variety of knowledge necessary to characterize a SASO scenario, but fail to bridge the gap between knowledge representation and COAA wargaming.

DARPA's ongoing Rapid Knowledge Formation (RKF) Program shares with KAGES the research goals of enabling codification of knowledge by nontechnical subject matter experts. RKF differs in scope, including many research groups working on such aspects as natural language understanding and knowledge discovery from text, as well as extensive reasoning on knowledge bases. While these objectives are largely outside the immediate scope of our project, RKF and its predecessor, the High Performance Knowledge Base (HPKB) project, have yielded valuable insight into issues of knowledge representation and engineering. Future research for KAGES will continue to consider the theoretical advances that these researchers contribute to the state of the art.

Another highly relevant area of research is the field of mixed-initiative planning, which is devoted to the study of software systems that can work in concert with the user to perform tasks that would be difficult for either one alone. These systems generally involve a collaborative planning component that plays a role similar to the KA Manager in the KAGES system; we have drawn heavily upon literature in this area as a source for useful algorithms.

James Allen (1996) at the University of Rochester has worked extensively on mixed-initiative planning, with a particular focus on collaborative planning viewed as a dialogue between agents (human or machine). His TRAINS system is an intelligent planning assistant that works with a human user to route freight trains. Karen Myers (1996) at SRI International has done much recent work in the area of "advice-taking" planners, which are based on the recognition that computers are better at managing the low-level complexities of the planning process, while human talents lie more towards more abstract plan guidance. Her system, the

Advisable Planner, is essentially a traditional generative planner (SIPE-2) fitted with a special interface allowing users to offer high-level strategic advice to steer the automated plan construction. Other notable research in the field includes the MI-CBP planner of Manuela Veloso (1997), at Carnegie Mellon, which is based upon a combination of the ForMAT case-based military planning system and the Prodigy automated planner, and Robert St. Amant's AIDE system (1996), which is a mixed-initiative planning system intended to assist a human with exploratory data analysis.

CONCLUSION

We have described here the architecture for a knowledge acquisition tool that will enable military analysts with no knowledge engineering experience to capture a wide range of SASO mission knowledge in a format suitable for use in wargaming simulations and decision aids. We have partially validated our approach through the construction of a limited prototype and through discussion with SASO subject matter experts, and we feel that the complete system will provide an innovative set of capabilities that could prove useful to the knowledge engineering and modeling communities.

The difficulty of the knowledge acquisition task is well-recognized, and many other research efforts have addressed the problem of making it less intimidating. The KAGES system we have outlined here does not solve this problem, but it does make certain inroads toward a solution. The application of existing mixed-initiative planning technology to the knowledge acquisition process is, we believe, thus far unique, and makes possible a new class of intelligent modeling tools. KAGES' explicit separation of presentation modality from knowledge acquisition plans and its ability to adapt to the level of the user are also powerful new techniques for enabling programmer-free knowledge acquisition – new to the KA community, that is, though quite familiar to the builders of intelligent tutoring systems. In short, the KAGES research is focused less on the development of revolutionary new knowledge representation algorithms and more on the application of a number of proven technologies from heretofore unrelated areas to the knowledge acquisition problem. The resulting synthesis shows promise as knowledge capture tool that can truly be said to be designed for domain experts rather than knowledge engineers.

AKNOWLEDGEMENT

REFERENCES

Ferguson, G., Allen, J.F., & Miller, B. (1996, May). TRAINS-95: towards a mixed-initiative planning assistant. In *Proceedings of the Third Int'l Conference on AI Planning Systems*, May 1996.

Hayes, C.C., & Schlabach, J.L. (1998). FOX-GA: a planning support tool for assisting military planners in a dynamic and uncertain environment. In AAAI Technical Report WS-98-02, AAAI Press.

Myers, K. L. (1996). Advisable planning systems. In Tate, A., *Advanced Planning Technology*. Menlo Park, CA: AAAI Press.

St. Amant, R., & Cohen, P. (1996). A planner for exploratory data analysis. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence.*

Veloso, M., Mulvehill, A., & Cox, M. (1997, July). Rationale-supported mixed-initiative case-based planning. In *Proceedings of IAAI-97, Innovative Applications of Artificial Intelligence*.